A Fuel-optimal Landing Guidance and Inverse Kinematics coupled PID Control Solution for Power-descent Vertical Landing in Simulation

Yongfeng Lu, Zejian Chen

Shenzhen Lokcol Interactive Ltd.

Block 8, 1st Weiye Road, Dongguan City, Guangdong Province, China

Email addresses: genxium@hotmail.com (Yongfeng Lu),
alteracfinished1111@gmail.com (Zejian Chen)

A Fuel-optimal Landing Guidance and Inverse Kinematics coupled PID Control Solution for Power-descent Vertical Landing in Simulation

Yongfeng Lu, Zejian Chen

Shenzhen Lokcol Interactive Ltd.

Block 8, 1^{st} Weiye Road, Dongguan City, Guangdong Province, China

Abstract

This article presents a guidance path planning and realtime control solution for power-descent landing of an ovoid-shape vehicle in simulation. Given a vehicle initially in free-fall state at a location in mid-air site, the proposed solution will guide and control it by the thrusters to land it at a target location on the ground.

The solution consists of an offline guidance path planning step and a realtime control step. It uses the result of a convexified guidance path planning to tune an *Inverse Kinematics (IK) coupled PID controller* with feedforward routes. In a *Bullet Physics*[1] based simulation environment, experiments were conducted to show good alignment with guidance, i.e. averaged *Root Mean Square Error (RMSE)* of position, velocity and attitude-in-quaternion are within $1.00\,m$, $2.00\,m/s$ and 0.10 respectively during a divert up to $120\,m$ horizontally and $100\,m$ vertically, against several disturbances(e.g. reducing mass, fluctuating *center of mass*, scheduling uncertainty and simulated wind), while the simulation frame rate is kept at around 60 fps for convenient realtime interaction.

Keywords: simulation, convex optimization, path planning, SOCP, Inverse Kinematics, PID, feedforward control, robust control

1. Introduction

Power-descent vertical landing has been of much interest in simulation of *Guidance, Navigation and Control(GNC)*. For *guidance* [2, 3] proposed a convex

Email addresses: genxium@hotmail.com (Yongfeng Lu), alteracfinished1111@gmail.com (Zejian Chen)

programming approach for planning a fuel-optimal path as well as the accompanying control sequence for the *center of mass*, which was brought to successful field test in cooperation with the vehicle and attitude controller provided by Masten Space Systems [4]. This guidance approach will be later reviewed and applied in section 4.1 for path planning with desired constraints.

While guidance approach takes care of path of the *center of mass*, for *control* an actual simulation of vehicle landing requires a *control* approach that applies to a 3D object with non-zero volume and an attitude. Recent literatures showed that several popular general purpose controllers, e.g. *Model Predictive Control (MPC)*[5, 6], *Linear Quadratic Regulator (LQR)*[7] and *PID* [7, 8, 9], are all feasible in their contexts.

Regardless of its popularity, *MPC* is not often an ideal choice when it comes to a need of fast response, i.e. in millisecond timescale [10]. Recent review[11] showed that in certain cases with parallelisation, iterative learning and other tailored optimisations, one can accelerate *MPC* significantly, yet still involving heavy empirical tuning. The *LQR control* on the otherhand, requires not only empirical tuning of the *Q, R* matrices but also of linearizing the dynamics by a chosen partial-state[7] when full-state linearization is not viable, which further complicates the design in theoretical analysis[12]. The traditional design methodology of *PID control* involves much empirical tuning for complicated dynamics as well, but can be analytical for certain simple dynamics[13, 14, 15].

This article proposed an *IK coupled PID controller* to help the vehicle follow the guidance path, where an *IK Estimator*(section 4.3) reduces the translational and attitude dynamics of interest to a simple *no delay, no damp double integrating transfer function* whose PID design parameters K_p, K_d can be analytically determined.

The simulation environment used by this article comes with more disturbances(section 5) than the aforementioned literatures as well, in terms of covering propellant consumption, varying *center of mass* of the vehicle, multithreading uncertainty and simulated wind, which the control should be robust to while keeping a high simulation frame rate.

2. Problem Description

Given a vehicle (a.k.a. the *plant*) initially at a position $\boldsymbol{r}_0$ and oriented with the *body frame* coordinates (X_b, Y_b, Z_b) all aligned with world frame coordinates (X_w, Y_w, Z_w) in free fall state, we'd try to land it to the target position $\boldsymbol{r}_{target}$ while minimizing propellant consumption during the flight, by maneuvering its thrusters. A successful landing is implied by *all 4 legs touching ground and vehicle speed is damped down to* $|\dot{\boldsymbol{r}}| < 10^{-2}$ *m/s*. An illustration of the coordinates when the vehicle is near r_{target} is presented by fig. 1.

Specifications of the *plant* are introduced below. All variables in this article are assumed *time varying* if not otherwise declared to be constant.

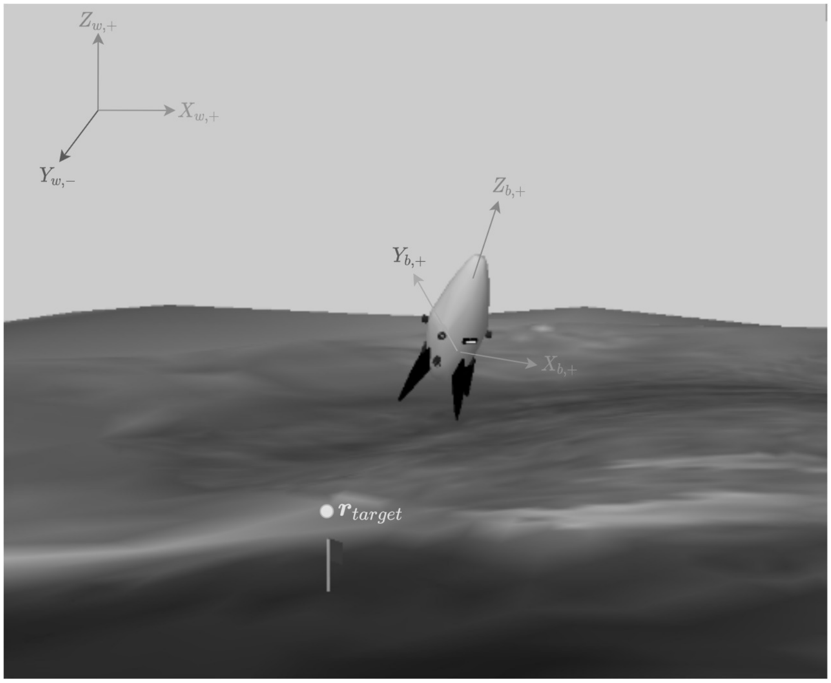


Figure 1: illustration of *body frame*, *world frame*, and r_{target}

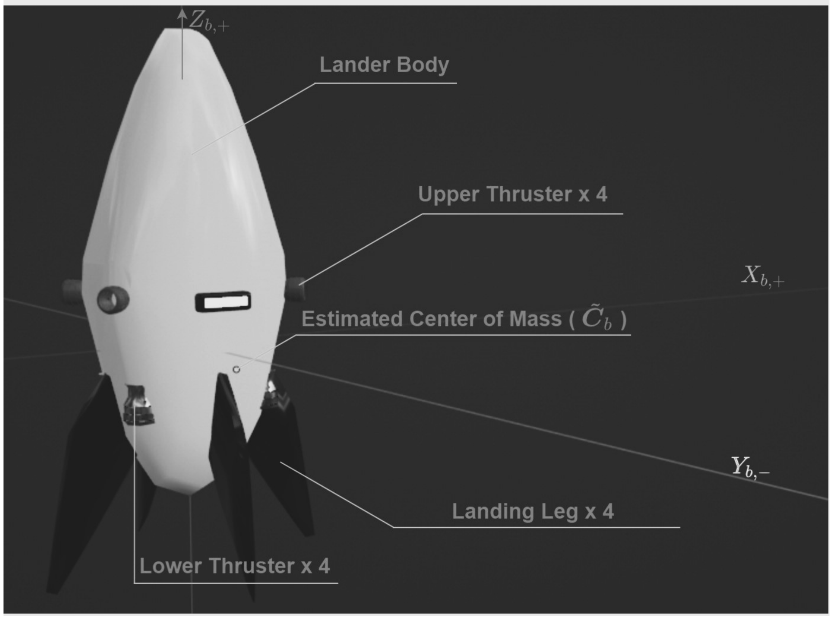


Figure 2: vehicle specs part 1

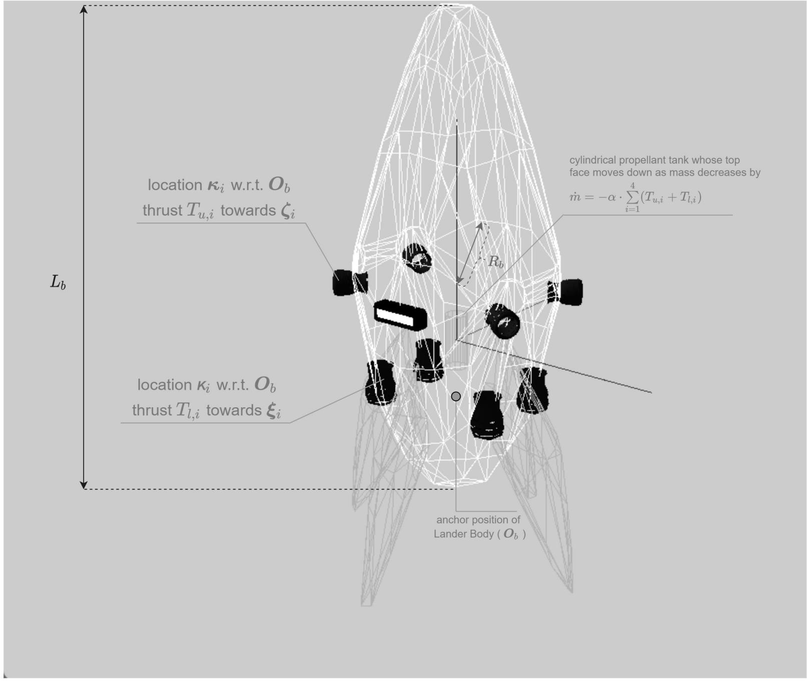

L_b
location $\boldsymbol{\kappa}_i$ w.r.t. $\boldsymbol{O}_b$
thrust $T_{u,i}$ towards $\boldsymbol{\zeta}_i$
cylindrical propellant tank whose top face moves down as mass decreases by
$\dot{m} = -\alpha \cdot \sum_{i=1}^{4} (T_{u,i} + T_{l,i})$
R_b
location $\boldsymbol{\kappa}_i$ w.r.t. $\boldsymbol{O}_b$
thrust $T_{l,i}$ towards $\boldsymbol{\xi}_i$
anchor position of Lander Body ($\boldsymbol{O}_b$)

Figure 3: vehicle specs part 2

- 1 ovoid shape *Lander Body* with constant anchor position $\boldsymbol{O}_b$ (the geometric center of mesh), dry mass m_b and dry inertia tensor $\boldsymbol{I}_{b,0}$ in *body frame*(i.e. always a diagonal matrix), major axis length L_b and largest cross-section radius R_b

- 4 massless *Upper Thrusters*, each located at $\boldsymbol{\kappa}_i$ w.r.t. $\boldsymbol{O}_b$, thrusting towards $\boldsymbol{\zeta}_i$ in *body frame* and with magnitude $T_{u,i} \in [\rho_1, \rho_2]$ & $\rho_1, \rho_2 > 0$ or being off, i.e. $T_{u,i} = 0$,

- 4 massless *Lower Thrusters*, each located at $\boldsymbol{\kappa}_i$ w.r.t. $\boldsymbol{O}_b$, thrusting towards $\boldsymbol{\xi}_i$ in *body frame* and with magnitude $T_{l,i} \in [\rho_1, \rho_2]$ & $\rho_1, \rho_2 > 0$ or being off, i.e. $T_{l,i} = 0$,

- 1 cylindrical propellant tank (simplified to include fuel and oxidizer) inside the body that initially contributes to some of the total mass, when flying the propellant is consumed by a rate proportional to the thrusts, i.e. $\dot{m} = -\alpha \cdot \sum\limits_{i=1}^{4} (T_{u,i} + T_{l,i})$ where m is the total mass of the whole lander; the decrement of m changes the overall *center of mass* during flight as well; the initial propellant mass is a constant $m_{prop,0}$

- 4 massless legs attached to the *Lander Body* with revolute joints to provide support when landed, each leg is also shaped with a considerable air contacting area to favor disturbance from section 5.3

- the rate of change in thrust magnitude of each thruster is unlimited

The navigation feedbacks of the *plant* in *world frame* are listed below.

a.1 $\begin{pmatrix}\boldsymbol{r}\\ \dot{\boldsymbol{r}}\end{pmatrix}$, position and velocity of the lander

a.2 $\begin{pmatrix}\boldsymbol{q}\\ \dot{\boldsymbol{q}}\end{pmatrix}$, attitude of the lander is represented by a quaternion $\boldsymbol{q}$ transforms $Z_{w,+}$ to align with the instantaneous $Z_{b,+}$ in fig. 1, while $\dot{\boldsymbol{q}}$ is derived from the measured world frame angular velocity $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$ by

$$\dot{\boldsymbol{q}} = \frac{\boldsymbol{W} \cdot \boldsymbol{q}}{2}$$

where

$\boldsymbol{W} = 0 + \omega_x \cdot \boldsymbol{i} + \omega_y \cdot \boldsymbol{j} + \omega_z \cdot \boldsymbol{k}$; ω_x is the angular speed around $X_{w,+}$ in world frame (see fig. 1), and similar arguments apply to ω_y, ω_z

a.3 m, total mass of the whole lander

The state-space measurement $\boldsymbol{s}$ and maneuver $\boldsymbol{\Xi}$ of this *plant* are denoted as follows.

$$\boldsymbol{s} := \begin{pmatrix}\boldsymbol{r}\\ \dot{\boldsymbol{r}}\\ \boldsymbol{q}\\ \dot{\boldsymbol{q}}\\ m\end{pmatrix} \tag{1}$$

$$\boldsymbol{\Xi} := \begin{pmatrix}\boldsymbol{T}_u\\ \boldsymbol{T}_l\end{pmatrix} \tag{2}$$

3. Controller Estimations

In fig. 4 the *IK Estimator* will take the following estimations to simplify the model.

b.1 $\tilde{\boldsymbol{C}}_b$, estimated *center of mass* of the whole lander, which is a constant position in *body frame*

b.2 $\tilde{\boldsymbol{I}}_b = \frac{m}{m_b} \cdot \boldsymbol{I}_{b,0}$, estimated inertia tensor of the whole lander w.r.t. $\tilde{\boldsymbol{C}}_b$ in *body frame*

The $\tilde{}$ on (**b.1**) and (**b.2**) means that it's an estimation by the controller.

Therefore the controller estimated dynamics in *body frame* would be

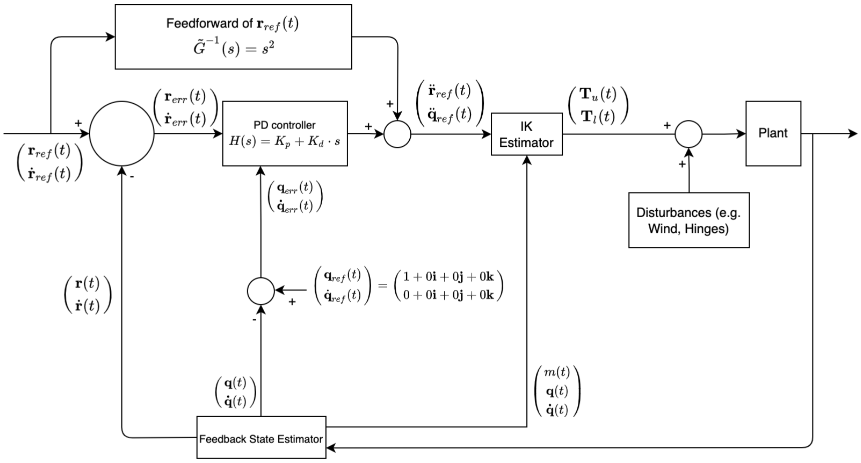

Feedforward of $\mathbf{r}_{ref}(t)$
$\tilde{G}^{-1}(s) = s^2$
+
$\begin{pmatrix}\mathbf{r}_{err}(t)\\ \dot{\mathbf{r}}_{err}(t)\end{pmatrix}$
PD controller
$H(s) = K_p + K_d \cdot s$
+
+
$\begin{pmatrix}\ddot{\mathbf{r}}_{ref}(t)\\ \ddot{\mathbf{q}}_{ref}(t)\end{pmatrix}$
IK
Estimator
$\begin{pmatrix}\mathbf{T}_u(t)\\ \mathbf{T}_l(t)\end{pmatrix}$
+
Plant
+
$\begin{pmatrix}\mathbf{r}_{ref}(t)\\ \dot{\mathbf{r}}_{ref}(t)\end{pmatrix}$
-
$\begin{pmatrix}\mathbf{q}_{err}(t)\\ \dot{\mathbf{q}}_{err}(t)\end{pmatrix}$
Disturbances (e.g.
Wind, Hinges)
$\begin{pmatrix}\mathbf{r}(t)\\ \dot{\mathbf{r}}(t)\end{pmatrix}$
$\begin{pmatrix}\mathbf{q}_{ref}(t)\\ \dot{\mathbf{q}}_{ref}(t)\end{pmatrix} = \begin{pmatrix}1 + 0\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}\\ 0 + 0\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}\end{pmatrix}$
+
-
$\begin{pmatrix}m(t)\\ \mathbf{q}(t)\\ \dot{\mathbf{q}}(t)\end{pmatrix}$
$\begin{pmatrix}\mathbf{q}(t)\\ \dot{\mathbf{q}}(t)\end{pmatrix}$
Feedback State Estimator

Figure 4: Block diagram of controller

$$\sum_{i=1}^{4} (T_{u,i} \cdot \boldsymbol{\zeta}_i + T_{l,i} \cdot \boldsymbol{\xi}_i) = \tilde{\boldsymbol{F}}_b \tag{3}$$

$$\sum_{i=1}^{4} (\boldsymbol{\kappa}_i \times (T_{u,i} \cdot \boldsymbol{\zeta}_i) + \mathbf{\kappa}_i \times (T_{l,i} \cdot \boldsymbol{\xi}_i)) = \tilde{\boldsymbol{M}}_b \tag{4}$$

where

$$\tilde{\boldsymbol{F}}_b := \boldsymbol{q}^{-1}((\ddot{\boldsymbol{r}} - \boldsymbol{g}) \cdot m)\boldsymbol{q} \tag{5}$$

$$\tilde{\boldsymbol{M}}_b := \tilde{\boldsymbol{I}}_b \cdot \tilde{\boldsymbol{\beta}}_b - \tilde{\boldsymbol{\omega}}_b \times (\tilde{\boldsymbol{I}}_b \cdot \tilde{\boldsymbol{\omega}}_b) \tag{6}$$

$$\tilde{\boldsymbol{\beta}}_b := \boldsymbol{q}^{-1} \cdot \dot{\boldsymbol{\omega}} \cdot \boldsymbol{q} \tag{7}$$

$$\tilde{\boldsymbol{\omega}}_b := \boldsymbol{q}^{-1} \cdot \boldsymbol{\omega} \cdot \boldsymbol{q} \tag{8}$$

Wind dynamics will only be taken as *disturbance*(section 5.3) as we don't have controllable fins in this problem.

4. Guidance and Control

4.1. Guidance planning

To land a vehicle from $\boldsymbol{r}_0$ to $\boldsymbol{r}_{target}$ with the hardware constraints given by section 2, we'd first employ a guidance algorithm to find a planned path $(\boldsymbol{r}_{ref}, \dot{\boldsymbol{r}}_{ref})$ for the vehicle to follow, while trying to minimize the total propellant consumption.

The guidance algorithm here concerns only the trajectory of *center of mass*, denote that

$$\tilde{\boldsymbol{F}} = \boldsymbol{q} \cdot \tilde{\boldsymbol{F}}_b \cdot \boldsymbol{q}^{-1} \tag{9}$$

, we can rephrase the problem with fuel optimal goal to the following. Kindly note that (9) is indeed a reverse of (5).

Problem 1 (non-convex)

$\max\limits_{t_f, \tilde{\boldsymbol{F}}} \tilde{m}(t_f) = \min\limits_{t_f, \tilde{\boldsymbol{F}}} \int_0^{t_f} \|\tilde{\boldsymbol{F}}\| \cdot dt$

where

$$\tilde{m}(0) = m_b + m_{prop,0}$$

$$\dot{\tilde{m}} = -\alpha \cdot \|\tilde{\boldsymbol{F}}\|$$

$$\tilde{\rho}_1 \leq \|\tilde{\boldsymbol{F}}\| \leq \tilde{\rho}_2$$

$$\dot{\boldsymbol{r}}(t_f) = \dot{\boldsymbol{r}}(0) = \boldsymbol{0}$$

$$\boldsymbol{r}(0) = \boldsymbol{r}_0$$

$$\boldsymbol{r}(t_f) = \boldsymbol{r}_{target}$$

$$\ddot{\boldsymbol{r}} = \frac{\tilde{\boldsymbol{F}}}{\tilde{m}} + \boldsymbol{g}$$

$$\tilde{m}(t_f) \geq m_b + (1 - \lambda) \cdot m_{prop,0}$$

$$r_z \geq r_{target,z}$$

Kindly note there's an empirically chosen constant λ, and an estimation $\tilde{m}$ instead of m is used here because in general $\dot{\tilde{m}} \leq \dot{m}$, thus we can be more conservative to constrain that $\lambda \leq 0.7$, i.e. plan to use only up to 70% of the propellant.

The new thrust bounds $\tilde{\rho}_1$, $\tilde{\rho}_2$ could've been estimated from ρ_1, ρ_2, $\rho_{side,1}$, $\rho_{side,2}$, Φ_{min}, Φ_{max} analytically, but in practice it's fair to just take the following approximations because we expect the main engine to contribute most to the thrust magnitude.

$$\tilde{\rho}_1 = \rho_1, \ \tilde{\rho}_2 = \rho_2$$

According to [2, 3] *Problem 1* is non-convex but can be losslessly convexified to the following *Second Order Cone Programming (SOCP)*[16].

Problem 2 (SOCP)

$\min\limits_{t_f, \boldsymbol{u}, \sigma} \int_0^{t_f} \sigma \cdot dt$

where

$$\eta(0) = \ln(m_b + m_{prop,0}) \tag{10}$$

$$\dot{\boldsymbol{r}}(t_f) = \dot{\boldsymbol{r}}(0) = \boldsymbol{0} \tag{11}$$

$$\boldsymbol{r}(0) = \boldsymbol{r}_0 \tag{12}$$

$$\boldsymbol{r}(t_f) = \boldsymbol{r}_{target} \tag{13}$$

$$\ddot{\boldsymbol{r}} = \boldsymbol{u} + \boldsymbol{g} \tag{14}$$

$$\mu_1(1 - (\eta - \eta_0) + \frac{(\eta - \eta_0)^2}{2}) \leq \sigma \leq \mu_2(1 - (\eta - \eta_0)) \tag{15}$$

$$\eta_0 \leq \eta \leq \ln(m_b + m_{prop,0} - \alpha \cdot \tilde{\rho}_1 \cdot t) \tag{16}$$

$$\|\boldsymbol{u}\| \leq \sigma \tag{17}$$

$$\dot{\eta} = -\alpha \cdot \sigma \tag{18}$$

$$\eta(t_f) \geq \ln(m_b + (1 - \lambda) \cdot m_{prop,0}) \tag{19}$$

$$r_z \geq r_{target,z} \tag{20}$$

by introducing new variables

$$\boldsymbol{u} := \frac{\tilde{\boldsymbol{F}}}{\tilde{m}}$$

$$\eta := \ln(\tilde{m})$$

$$\eta_0 := \ln(m_b + m_{prop,0} - \alpha \cdot \tilde{\rho}_2 \cdot t)$$

$$\mu_1 := \tilde{\rho}_1 \cdot e^{-\eta_0}, \mu_2 = \tilde{\rho}_2 \cdot e^{-\eta_0}$$

The following *low speed* constraints can be added to *Problem 2* without breaking the SOCP compliance.

$$\dot{\boldsymbol{r}}_z \leq given\, descending\, speed\, limit \tag{21}$$

$$|\ddot{\boldsymbol{r}}_z| \leq given\, descending\, acceleration\, limit \tag{22}$$

When given a certain termination time-of-flight t_f and discretized about t, *Problem 2* can be numerically solved in an efficient manner by using SOCP specific solvers like *Embedded Conic Solver (ECOS)*[17] in search for a feasible t_f will be in range $t_f \in [\frac{m_b \cdot |\dot{\boldsymbol{r}}(0)|}{\rho_2}, \frac{m_{prop,0}}{\rho_1}]$.

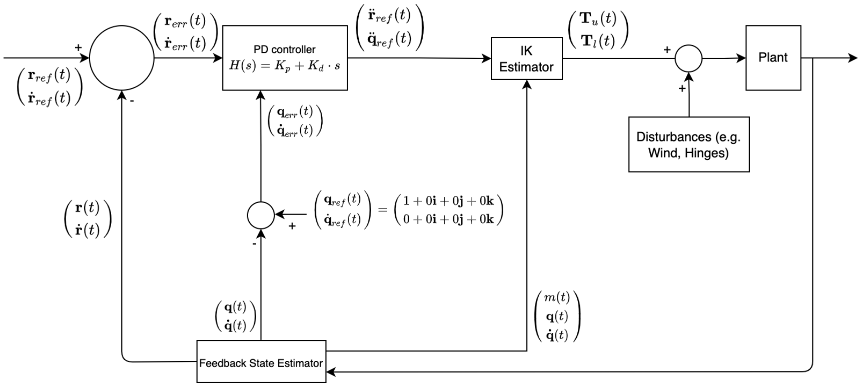
4.2. Controller design

Given the guidance path $\begin{pmatrix} \boldsymbol{r}_{ref} \\ \dot{\boldsymbol{r}}_{ref} \end{pmatrix}$, the vehicle needs a control algorithm to actually follow the path for landing. The control algorithm aims to find proper thrusts $\boldsymbol{T}_u, \boldsymbol{T}_l$ whenever the controller is active, and these thrusts are employed by the thrusters to apply dynamics to the vehicle.

The (3), (4) w.r.t. (1) & (2) form a *Multi Input Multi Output (MIMO)* system, an intuitive choice would be to obtain a linearized $\dot{\boldsymbol{q}} = \boldsymbol{A} \cdot \boldsymbol{q} + \boldsymbol{B} \cdot \boldsymbol{w}$ from (3), (4) at a chosen trim point, then apply a Model Predictive Control by scoring proximity to $\begin{pmatrix} \boldsymbol{r}_{ref} \\ \dot{\boldsymbol{r}}_{ref} \end{pmatrix}$.

However there's inconvenience for linearizing this model. Only part of $\boldsymbol{q}$ can be linearized because $\dot{m} \neq 0$ during flight, and a non-hover flight might remove $\begin{pmatrix} \boldsymbol{r} \\ \dot{\boldsymbol{r}} \end{pmatrix}$ from linearizable parts over all time[7], resulting in extra complications in design of the whole controller as the non-linearized parts are not decoupled.

For this problem the relatively simple PID controllers with the non-linear model suffices to work to satisfaction. By decoupling inputs and outputs into several *Single-Input-Single-Output (SISO)* PID controllers, we avoid empirically guessing any cross coupled factors for (2) w.r.t. (**a.1**), (**a.2**), (**a.3**), as shown in the figure below.


+
$\begin{pmatrix} \mathbf{r}_{ref}(t) \\ \dot{\mathbf{r}}_{ref}(t) \end{pmatrix}$
-
$\begin{pmatrix} \mathbf{r}_{err}(t) \\ \dot{\mathbf{r}}_{err}(t) \end{pmatrix}$
PD controller
$H(s) = K_p + K_d \cdot s$
$\begin{pmatrix} \ddot{\mathbf{r}}_{ref}(t) \\ \ddot{\mathbf{q}}_{ref}(t) \end{pmatrix}$
IK Estimator
$\begin{pmatrix} \mathbf{T}_u(t) \\ \mathbf{T}_l(t) \end{pmatrix}$
+
+
Plant
Disturbances (e.g. Wind, Hinges)
$\begin{pmatrix} \mathbf{q}_{err}(t) \\ \dot{\mathbf{q}}_{err}(t) \end{pmatrix}$
$\begin{pmatrix} \mathbf{r}(t) \\ \dot{\mathbf{r}}(t) \end{pmatrix}$
$\begin{pmatrix} \mathbf{q}_{ref}(t) \\ \dot{\mathbf{q}}_{ref}(t) \end{pmatrix} = \begin{pmatrix} 1 + 0\mathbf{i} + 0\mathbf{j} + 0\mathbf{k} \\ 0 + 0\mathbf{i} + 0\mathbf{j} + 0\mathbf{k} \end{pmatrix}$
+
-
$\begin{pmatrix} \mathbf{q}(t) \\ \dot{\mathbf{q}}(t) \end{pmatrix}$
$\begin{pmatrix} m(t) \\ \mathbf{q}(t) \\ \dot{\mathbf{q}}(t) \end{pmatrix}$
Feedback State Estimator

* in fig. 4 a feedforward route $\tilde{G}^{-1}(s) = s^2$ which reduces delay in $\dot{\boldsymbol{r}}$ tracking $\dot{\boldsymbol{r}}_{ref}$[18] is added

Figure 5: Block diagram of non-feedforward controller

First, each of the SISO PID controllers would seek the proper $\boldsymbol{r}_{ref}(t_{ref})$ as the instantaneous reference location to track by KD-tree search[19], and is just a classical PD controller responsible for position or attitude control

$$\tilde{G}(s) = \frac{1}{s^2} \tag{23}$$

$$H_{\boldsymbol{r}}(s) = K_{p,\boldsymbol{r}} + s \cdot K_{d,\boldsymbol{r}} \tag{24}$$

$$H_{\boldsymbol{q}}(s) = K_{p,\boldsymbol{q}} + s \cdot K_{d,\boldsymbol{q}} \tag{25}$$

$$\mathcal{L}\{\boldsymbol{r}\} = \frac{\tilde{G} \cdot H_{\boldsymbol{r}}}{1 + \tilde{G} \cdot H_{\boldsymbol{r}}} \cdot \mathcal{L}\{\boldsymbol{r}_{ref}\} \tag{26}$$

$$\mathcal{L}\{\boldsymbol{q}\} = \frac{\tilde{G} \cdot H_{\boldsymbol{q}}}{1 + \tilde{G} \cdot H_{\boldsymbol{q}}} \cdot \mathcal{L}\{\boldsymbol{q}_{ref}\} \tag{27}$$

where $\mathcal{L}\{\cdot\}$ indicates the Laplace transform.

The ˜ on (23) means that it's an estimation of what the *IK Estimator* and *Plant* steps together would approximately achieve

$$\begin{pmatrix} \boldsymbol{r}(t) \\ \boldsymbol{q}(t) \end{pmatrix} = \int_0^t \int_0^t \begin{pmatrix} \ddot{\boldsymbol{r}}_{ref}(\tau) \\ \ddot{\boldsymbol{q}}_{ref}(\tau) \end{pmatrix} \cdot d\tau^2$$

in good fidelity.

That (23) being a *no delay, no damp double integrating* transfer function makes it a peculiar second order system to design the PID controller. In a wellknown case

$$\tilde{G}_{alt}(s) = \frac{e^{-\gamma s}}{(\delta_1 s + 1)(\delta_2 s + 1)} \tag{28}$$

, simple IMC approach to estimate 2^{nd}-order approximator and then a single parameter tunable PID controller is proposed in [13]. However in a simulation environment without air dynamics, (23) will remain as-is and thus even the treatments in [14, 15] wouldn't be applicable due to $\gamma = 0$.

Aiming at an analytical approach to determine the variables in (24), (25) and knowing that derivative action is necessary for stabilization for (23)[14], every PID controller is limited to be a PD controller in our approach and the steady state error will instead be complemented by a feedforward route.

To compute proper $K_{p,\boldsymbol{r}}$ and $K_{d,\boldsymbol{r}}$, as we're also constrained by $T_{settle} < t_f$ from the section 4.1, an analytical $\tilde{T}_{settle}(K_p, K_d)$ for each dimension is preferable given requirements of the unit step response of (26)

c.1 λ_{settle}, the settled response amplitude ratio threshold

c.2 pOS_0, the expected peak overshoot

c.3 $\tilde{T}_{settle,0} = (0.6 \cdot t_f)$, the intersection of amplitude envelope of $\mathcal{L}^{-1}\{\frac{\tilde{G} \cdot H_{\boldsymbol{r}}}{1+\tilde{G} \cdot H_{\boldsymbol{r}}} \cdot \frac{1}{s}\}$ and the *SettlingTimeThreshold* horizon

, to numerically find a fit for both

d.1 $|\frac{\tilde{T}_{settle}(K_{p,\boldsymbol{r}}, K_{d,\boldsymbol{r}})}{\tilde{T}_{settle,0}} - 1| < 0.05$, and

d.2 $|\frac{pOS(K_{p,\boldsymbol{r}}, K_{d,\boldsymbol{r}})}{pOS_0} - 1| < 0.05$

, where the analytical form of $\tilde{T}_{settle}(\cdot, \cdot)$ and $pOS(\cdot, \cdot)$ can be calculated by symbolic calculation tools.

Next, given $\begin{pmatrix} \ddot{\boldsymbol{r}}_{ref} \\ \ddot{\boldsymbol{q}}_{ref} \end{pmatrix}$ computed by the *PD Controller* step, the *IK Estimator* will compute computes the instantaneous (2) numerically in *body frame* for maneuver. Details will be described in section 4.3.

Lastly, the *Plant* in simulation will guarantee to clip $T_{l,i} \in [\rho_1, \rho_2]$ and $T_{u,i} \in [\rho_1, \rho_2]$ before stepping in the physics engine.

4.3. IK Estimator

Following the PID controller output, the proposed *IK Estimator* computes

$$\dot{\boldsymbol{\omega}} = 2 \cdot (\ddot{\boldsymbol{q}} \cdot \boldsymbol{q} - (\dot{\boldsymbol{q}} \cdot \boldsymbol{q}^{-1})^2) \tag{29}$$

, and then the desired instantaneous $\tilde{\boldsymbol{F}}_b$ and $\tilde{\boldsymbol{M}}_b$ by (5), (6).

Eventually (2) is numerically solved to approximately fulfill (3), (4) as a minimization problem

Problem 3 (thrust lower bound loosen)

$$\min_{\boldsymbol{T}'_u, \boldsymbol{T}'_l} \left| \begin{pmatrix} \sum_{i=1}^{4} (T'_{u,i} \cdot \boldsymbol{\zeta}_i + T'_{l,i} \cdot \boldsymbol{\xi}_i) - \tilde{\boldsymbol{F}}_b \\ \sum_{i=1}^{4} (\boldsymbol{\kappa}_i \times (T'_{u,i} \cdot \boldsymbol{\zeta}_i) + \boldsymbol{\kappa}_i \times (T'_{l,i} \cdot \boldsymbol{\xi}_i)) - \tilde{\boldsymbol{M}}_b \end{pmatrix} \right| \tag{30}$$

for constraints

$$\begin{aligned} T'_{u,i} &\in [0, \rho_2] \\ T'_{l,i} &\in [0, \rho_2] \end{aligned}$$

$$\sum_{i=1}^{4} (T'_{u,i} \cdot \boldsymbol{\zeta}_i + T'_{l,i} \cdot \boldsymbol{\xi}_i) = \tilde{\boldsymbol{F}}_b \tag{3 revisited}$$

$$\sum_{i=1}^{4} (\boldsymbol{\kappa}_i \times (T'_{u,i} \cdot \boldsymbol{\zeta}_i) + \boldsymbol{\kappa}_i \times (T'_{l,i} \cdot \boldsymbol{\xi}_i)) = \tilde{\boldsymbol{M}}_b \tag{4 revisited}$$

The overdetermined linear system (3), (4) could be violated to a certain extent at some instant of time, thus *Problem 3* is solved by *Sequential Least Squares Programming (SLSQP)*[20] which can allow violation of constraints to certain extent for engineering practice.

For any solution $\boldsymbol{T}'_u, \boldsymbol{T}'_l$ to *Problem 3*, each value $T'_{u,i}$, $T'_{l,i}$ will be clipped into $[\rho_1, \rho_2]$ or 0 by the simulation thread to guarantee that thrust magnitude constraints are never violated.

After clipping $\boldsymbol{T}'_u, \boldsymbol{T}'_l$ into $\boldsymbol{T}_u, \boldsymbol{T}_l$ ($T_{u,i} \in [\rho_1, \rho_2]$ or $T_{u,i} = 0$, $T_{l,i} \in [\rho_1, \rho_2]$ or $T_{l,i} = 0$), the violation is chosen to be recorded by

$$|\boldsymbol{Q}| := \left| \begin{pmatrix} \sum_{i=1}^{4} (T_{u,i} \cdot \boldsymbol{\zeta}_i + T_{l,i} \cdot \boldsymbol{\xi}_i) - \tilde{\boldsymbol{F}}_b \\ (\sum_{i=1}^{4} (\boldsymbol{\kappa}_i \times (T_{u,i} \cdot \boldsymbol{\zeta}_i) + \boldsymbol{\kappa}_i \times (T_{l,i} \cdot \boldsymbol{\xi}_i)) - \tilde{\boldsymbol{M}}_b)/\mathcal{J} \end{pmatrix} \right| \tag{31}$$

where $\mathcal{J}\,(unit:\ m)$ is the normalizing factor that makes all dimensions of $\boldsymbol{Q}$ of same unit N. Discussion of $|\boldsymbol{Q}|$ will be presented in section 6.2.

5. Disturbances

5.1. Scheduling uncertainty

In runtime, a controller thread is scheduled independently from the physics simulation thread, where the physics simulation thread is responsible for applying the thrusts (determined by controller thread) as impulses during the discrete time step. Each thread gets uncertainty for the inter-tick duration as well as order of scheduling, thus resulting in different outputs for tests of same inputs.

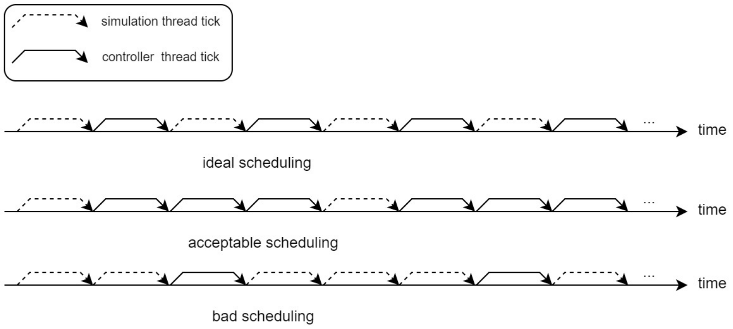


Figure 6: Different possible schedulings of simulation thread and controller thread

In section 6 the tests are run on a *non-Realtime Operating System*, where an unexpected order of scheduling could significantly degrade the performance of controlling. For example if the physics simulation thread ticks consecutively 3 times with an outdated set of thrusts without a new tick of controller thread to measure and update as shown in *bad scheduling* of fig. 6, the *plant* might become difficult to land by following the guidance path. The controller thread could be scheduled such that it ticks at a specified minimum frequency, e.g. 120 fps, in a deterministic manner if the tests were run on a *Realtime Operating System.*

The following records are kept for scheduling uncertainty.

e.1 *simCnt*, the accumulated count of physics simulation thread ticks

e.2 *ctrlCnt*, the accumulated count of controller thread ticks

e.3 $\mathcal{T} := 1 + sign(\frac{d(simCnt/ctrlCnt)}{dt})$, the time series for *bad scheduling* occurrences

Discussion of $\mathcal{T}$ will be presented in section 6.2.

5.2. Fluctuating center of mass and inertia

As shown in fig. 3 of section 2, the top face of the cylindrical propellant tank will be lowered as the propellant mass decreases, resulting in a time varying *center of mass* $\boldsymbol{C}_b$ and *inertia* $\boldsymbol{I}_b$ in *body frame.*

The fluctuating *center of mass* and *inertia* are accounted as eqs **b.1** and **b.2** respectively by the controller.

5.3. Simulated wind

The simulated wind is simplified as inviscid, incompressible and irrotational everywhere, imposing forces on the triangular mesh surface of the lander.

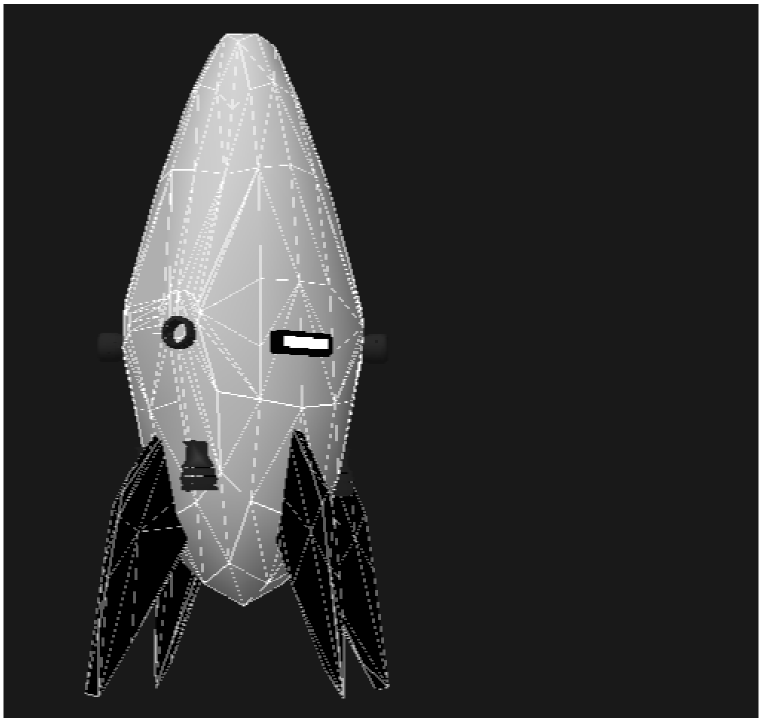


Figure 7: Triangular mesh surface of the lander

On each triangle surface with area S_i and norm $\boldsymbol{n}_i$, the forces imposed by the wind in *world frame* are

$$\boldsymbol{f}_{drag} = -0.6 \cdot \rho_{air} \cdot \boldsymbol{v}_d \cdot (\boldsymbol{S}_i \cdot \boldsymbol{v}_d)$$
$$\boldsymbol{f}_{lift} = \begin{cases} 0.4 \cdot \rho_{air} \cdot (\boldsymbol{S}_i \times \boldsymbol{v}_d \times \boldsymbol{n}_d), & \text{if } 0 < \boldsymbol{n}_d \cdot \boldsymbol{n}_i < 0.98 \\ \boldsymbol{0}, & \text{otherwise} \end{cases}$$

where

$$\boldsymbol{S}_i = S_i \boldsymbol{n}_i$$
$$\boldsymbol{v}_d = \dot{\boldsymbol{r}} - \boldsymbol{v}_{wind}$$
$$\boldsymbol{n}_d = \frac{\boldsymbol{v}_d}{|\boldsymbol{v}_d|}$$

The simulated wind is not accounted by the controller either.

6. Simulation Test

Simulation tests were conducted for 6 *input sets*, each *input set* has a unique combination of ($\textit{wind profile \#}$, r_0, r_{target}).

All *input sets* share the same configuration of constants in section 6.1. Each *input set* contains 20 *episodes* and all episodes under the same *input set* share the same ($\textit{wind profile \#}$, r_0, r_{target}) and thus the same computed guidance path $\begin{pmatrix} \boldsymbol{r}_{ref} \\ \dot{\boldsymbol{r}}_{ref} \end{pmatrix}$.

For each *input set*, 4 major performance indicators $\overline{\Delta}(\boldsymbol{r})$, $\overline{\Delta}(\dot{\boldsymbol{r}})$, $\overline{\Delta}(|\boldsymbol{q}|)$, $\overline{\Delta}(m)$ will be presented. Moreover, we examined several possible causes for the relatively poor alignment of m *v.s.* m_{ref} by discussing $Pr(\mathcal{T}, \frac{\dot{m}}{\dot{m}_{ref}})$ and $Pr(|\boldsymbol{Q}|, \frac{\dot{m}}{\dot{m}_{ref}})$.

Here Δ denotes *Root Mean Square Error (RMSE)* for the measured values is w.r.t. corresponding guidance value, and Pr denotes *Pearson's correlation coefficient*[21], and the $^{-}$ of each column in *Outputs* indicates that it's averaged from 20 episodes of the same inputs over $[0, \text{measured } t_f]$ *seconds*.

6.1. Constants

All using SI units, while dimensions and weights are referencing the Masten XL-1 profile.

f.1 $\boldsymbol{g} = \begin{pmatrix} 0 \\ 0 \\ -9.81 \end{pmatrix} m^2/s$, same gravity as the earth

f.2 $L_b = 4.424\,m$, $R_b = 0.8755\,m$, $\mathcal{J} = 1.0\,m$

f.3 $m_b = 728.28\,kg$, $m_{prop,0} = 1699.32\,kg$

f.4 $\rho_1 = 7282.8\,N$, $\rho_2 = 24276.0\,N$

f.5 $\boldsymbol{\kappa} = \begin{pmatrix} 0.86602 & -0.86602 & -0.86602 & 0.86602 \\ 0.5 & 0.5 & -0.5 & -0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 \end{pmatrix} m$

f.6 $\mathbf{\kappa} = \begin{pmatrix} 0.60622 & -0.60622 & -0.60622 & 0.60622 \\ 0.35 & 0.35 & -0.35 & -0.35 \\ -0.5 & -0.5 & -0.5 & -0.5 \end{pmatrix} m$

f.7 $\boldsymbol{\zeta} = \begin{pmatrix} cos\frac{\pi}{5} & cos\frac{4\pi}{5} & cos\frac{6\pi}{5} & cos\frac{9\pi}{5} \\ sin\frac{\pi}{5} & sin\frac{4\pi}{5} & sin\frac{6\pi}{5} & sin\frac{9\pi}{5} \\ 0 & 0 & 0 & 0 \end{pmatrix}$

f.8 $\boldsymbol{\xi} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}$

f.9 $\lambda_{settle} = 0.05$, $pOS_0 = 0.08$, $\tilde{T}_{settle,0} = 1.77\,s$, according to section 4.2 the controller parameters are $K_{p,\boldsymbol{r}} = K_{p,\boldsymbol{q}} = 3.00$, $K_{d,\boldsymbol{r}} = K_{d,\boldsymbol{q}} = 5.00$

f.10 guidance discretization fps = 15 frames/s

f.11 simulation & renderer fps = 60 frames/s

Table 1: Wind profiles

wind#	$\boldsymbol{v}_{wind}(\boldsymbol{r},t)\ m/s$	$\rho_{air}(\boldsymbol{r},t)\ kg/m^3$
#1	const. $\begin{pmatrix} 0.0 \\ 0 \\ 0 \end{pmatrix}$	const. 1.225
#2	const. $\begin{pmatrix} 3.5 \\ 0 \\ 0 \end{pmatrix}$	const. 1.225
#3	$\begin{pmatrix} 3.5 \\ 0 \\ 0 \end{pmatrix} \cdot log(\frac{r_z+20}{25})$	$1.225 \cdot (1 - \frac{0.0065 \cdot r_z}{288.16})^{4.2561}$

6.2. Results and Discussion

All episodes had their vehicles landed successfully. The measured results are good in terms of location, velocity and attitude alignment w.r.t. guidance, having $\overline{\Delta}(\boldsymbol{r}) < 1.00\,m$, $\overline{\Delta}(\dot{\boldsymbol{r}}) < 2.00\,m/s$ and $\overline{\Delta}(|\boldsymbol{q}|) < 0.10$ respectively during a divert up to $120\,m$ horizontally and $100\,m$ vertically.

The $\overline{\Delta}(m)$ of each *input set* implies a relatively poor m *v.s.* m_{ref} alignment. In some typical transient plots figs. 8 to 10 it can be seen that $\overline{\Delta}(\dot{m})$ is mostly aligned with guidance, yet spiked out of guidance for some short periods. Further investigation showed that both eq. (31) could be one of the causes for such misalignment as $\overline{Pr}(|\boldsymbol{Q}|, \frac{\dot{m}}{\dot{m}_{ref}}) \in [0.22, 0.77]$ and mostly lean on the larger side for all *input sets*. Similarly $\overline{Pr}(\mathcal{T}, \frac{\dot{m}}{\dot{m}_{ref}}) \in [0.30, 0.64]$ also implies that eq **e.3** could be one of the causes for this misalignment of mass consumption.

Another cause of the poor m *v.s.* m_{ref} alignment could be that the propellant consumption rate assumed by guidance is inherently inaccurate, as simulation runs with 8 independent thrusters to provide equivalent total force on the *center of mass* required by section 4.1.

7. Conclusion

The guided path can be tracked closely by proposed controller(section 4.2) regardless of that the rotational dynamics and disturbances(section 5) are not accounted in section 4.1. The approach here is deterministic, capable of not only landing the plant but also imposing practical constraints such as descending speed, gliding cone region etc. on the path.

Table 2: Test results

Inputs			Outputs				
wind#	$\boldsymbol{r}_0$, $\boldsymbol{r}_{target}$	Guidance t_f, Measured $\overline{t}_f$	$\overline{\Delta}(\boldsymbol{r})$, $\overline{\Delta}(\dot{\boldsymbol{r}})$ $\overline{\Delta}(\vert\boldsymbol{q}\vert)$	$\overline{\Delta}(m)$, $\overline{\Delta}(\dot{m})$	$\overline{Pr}(\mathcal{T}, \vert\boldsymbol{q}_{ref}^{-1}\boldsymbol{q}\vert)$, $\overline{Pr}(\mathcal{T}, \frac{\dot{m}}{\dot{m}_{ref}})$	$\overline{Pr}(\vert\boldsymbol{Q}\vert, \vert\boldsymbol{q}_{ref}^{-1}\boldsymbol{q}\vert)$, $\overline{Pr}(\vert\boldsymbol{Q}\vert, \frac{\dot{m}}{\dot{m}_{ref}})$	$\overline{Pr}(\vert\boldsymbol{Q}\vert, \mathcal{T})$
#1	$\begin{pmatrix}40.00\\40.00\\60.00\end{pmatrix},\begin{pmatrix}-40.00\\-10.00\\-1.18\end{pmatrix}$	13.00, 12.34	0.29, 0.70, 0.01	139.36, 109.80	0.22, 0.56	0.21, 0.37	0.43
#1	$\begin{pmatrix}40.00\\80.00\\100.00\end{pmatrix},\begin{pmatrix}-40.00\\-10.00\\-1.18\end{pmatrix}$	14.00, 14.57	0.54, 1.78, 0.04	206.53, 184.79	0.10, 0.32	0.31, 0.77	0.13
#2	$\begin{pmatrix}40.00\\40.00\\60.00\end{pmatrix},\begin{pmatrix}-40.00\\-10.00\\-1.18\end{pmatrix}$	13.00, 12.23	0.27, 0.61, 0.01	126.88, 100.55	0.08, 0.64	0.17, 0.37	0.50
#2	$\begin{pmatrix}40.00\\80.00\\100.00\end{pmatrix},\begin{pmatrix}-40.00\\-10.00\\-1.18\end{pmatrix}$	14.00, 14.33	0.57, 1.74, 0.05	206.01, 166.20	0.04, 0.30	0.36, 0.76	0.76
#3	$\begin{pmatrix}40.00\\40.00\\60.00\end{pmatrix},\begin{pmatrix}-40.00\\-10.00\\-1.18\end{pmatrix}$	13.00, 12.24	0.29, 0.65, 0.02	133.55, 99.35	0.09, 0.51	0.25, 0.22	0.37
#3	$\begin{pmatrix}40.00\\80.00\\100.00\end{pmatrix},\begin{pmatrix}-40.00\\-10.00\\-1.18\end{pmatrix}$	14.00, 14.13	0.59, 1.60, 0.04	187.49, 148.19	-0.03, 0.36	0.19, 0.70	0.10

In reality, engine thrust throttling capability is usually limited, i.e. with extra propellant loss when re-ignited or not re-ignitable, or could only reach certain values in $[\rho_1, \rho_2]$ [22, 23], thus further development of the approach should take re-ignition cost and limitations, as well as discrete thrust throttling levels into account. Controllable fins and gimbaled thrusters can be considered to mitigate control difficulty introduced by limited thrust throttling capability, which could be included in further development in section 4.2 to take the non-linear dynamics and wind disturbance into account.

References

[1] E. Coumans, Bullet physics simulation, in: ACM SIGGRAPH 2015 Courses, SIGGRAPH '15, Association for Computing Machinery, New York, NY, USA, 2015. `doi:10.1145/2776880.2792704`.
URL `https://doi.org/10.1145/2776880.2792704`

[2] B. Acikmese, S. R. Ploen, Convex programming approach to powered descent guidance for mars landing, Journal of Guidance, Control, and Dynamics 30 (5) (2007) 1353–1366.

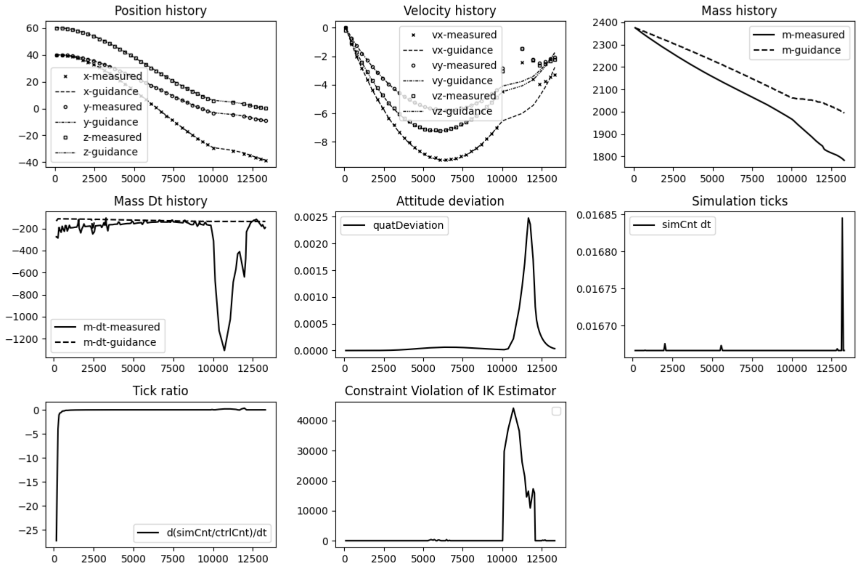


Figure 8: Episode 15 of 1^{st} input set

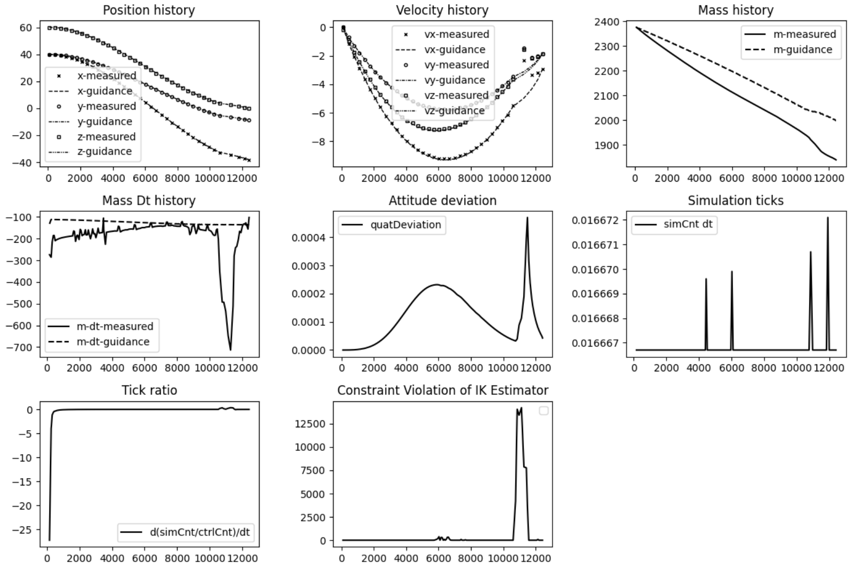


Figure 9: Episode 2 of 5^{th} input set

[3] B. Açıkmeşe, J. M. Carson, L. Blackmore, Lossless convexification of non-convex control bound and pointing constraints of the soft landing optimal

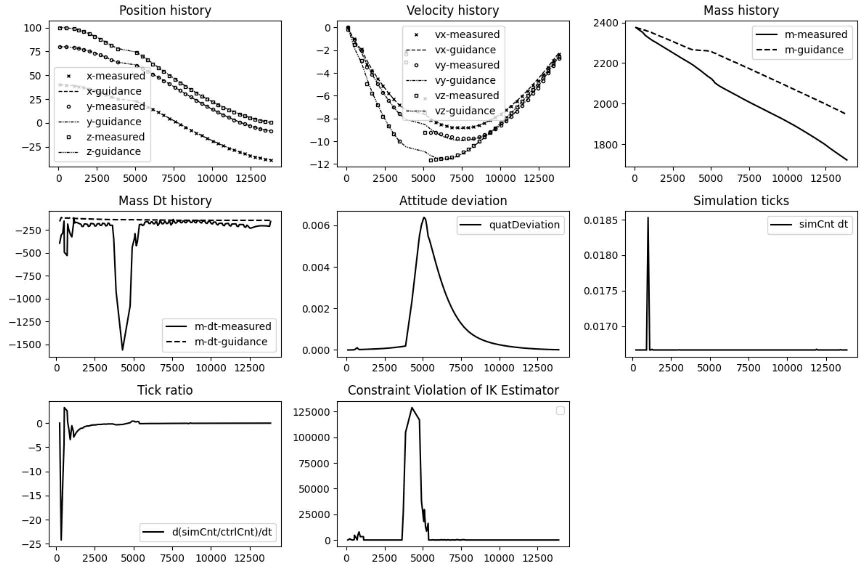


Figure 10: Episode 4 of 6^{th} input set

control problem, IEEE Transactions on Control Systems Technology 21 (6) (2013) 2104–2113.

[4] B. Acikmese, J. Casoliva, J. Carson, L. Blackmore, G-fold: A real-time implementable fuel optimal large divert guidance algorithm for planetary pinpoint landing, Concepts and Approaches for Mars Exploration 1679 (2012) 4193.

[5] J. Wang, N. Cui, C. Wei, Optimal rocket landing guidance using convex optimization and model predictive control, Journal of guidance, control, and dynamics 42 (5) (2019) 1078–1092.

[6] A. Botelho, M. Martinez, C. Recupero, A. Fabrizi, G. De Zaiacomo, Design of the landing guidance for the retro-propulsive vertical landing of a reusable rocket stage, CEAS Space Journal (2022) 1–14.

[7] S. Kim, N. Das, R. Bhattacharya, Modeling and optimal control of hybrid uavs with wind disturbance (2020). `arXiv:2006.11192`.

[8] S.-J. Jo, C.-O. Min, D.-W. Lee, K.-R. Cho, Control of powered descent phase for a lunar lander using pid controller, Journal of the Korean Society for Aeronautical & Space Sciences 39 (5) (2011) 408–415.

[9] Z. Hilmi, H. ŞENOL, F. GÜNER, Lunar excursion module landing control system design with p, pi and pid controllers, Karadeniz Fen Bilimleri Dergisi 9 (2) (2019) 390–405.

[10] H. J. Ferreau, Model predictive control algorithms for applications with millisecond timescales, Arenberg Doctoral School of Science, Engineering & Technology, Faculty of Engineering, Department of Electrical Engineering (2011).

[11] M. Schwenzer, M. Ay, T. Bergs, D. Abel, Review on model predictive control: an engineering perspective, The International Journal of Advanced Manufacturing Technology 117 (5) (2021) 1327–1349.

[12] I. Sarras, A. Venkatraman, R. Ortega, A. van der Schaft, Partial linearization of mechanical systems with application to observer design, Citeseer, 2008.

[13] S. Skogestad, C. Grimholt, The simc method for smooth pid controller tuning, in: PID control in the third millennium, Springer, 2012, pp. 147–175.

[14] C. Grimholt, S. Skogestad, Optimal pid control of double integrating processes, IFAC-PapersOnLine 49 (7) (2016) 127–132.

[15] D. L. D. Ruscio, C. Dalen, Tuning pd and pid controllers for double integrating plus time delay systems (2017).

[16] F. Alizadeh, D. Goldfarb, Second-order cone programming, Mathematical programming 95 (1) (2003) 3–51.

[17] A. Domahidi, E. Chu, S. Boyd, Ecos: An socp solver for embedded systems, in: 2013 European Control Conference (ECC), IEEE, 2013, pp. 3071–3076.

[18] A. Visioli, A new design for a pid plus feedforward controller, Journal of Process Control 14 (4) (2004) 457–463.

[19] S. Maneewongvatana, D. M. Mount, Analysis of approximate nearest neighbor searching with clustered point sets, arXiv preprint cs/9901013 (1999).

[20] D. Kraft, Algorithm 733: Tomp–fortran modules for optimal control calculations, ACM Transactions on Mathematical Software (TOMS) 20 (3) (1994) 262–281.

[21] Student, Probable error of a correlation coefficient, Biometrika (1908) 302–310.

[22] M. J. Casiano, J. R. Hulka, V. Yang, Liquid-propellant rocket engine throttling: A comprehensive review, Journal of propulsion and power 26 (5) (2010) 897–923.

[23] E. Betts, R. Frederick, A historical systems study of liquid rocket engine throttling capabilities, in: 46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, 2010, p. 6541.